UNCLASSIFIED

Registration No.

21207

-Technical Report-

Triangular Mesh Generation of Elevation Versus Distance Course Profiles for use with Road Property Files

DISTRIBUTION STATEMENT A.
Approved For Public Release; Distribution Unlimited

September 2010

U.S. Army Tank Automotive Research, Development, and Engineering Center
Detroit Arsenal
Warren, Michigan 48397-5000

UNCLASSIFIED

REPORT DOCUMENTATION PAGE			*Form Approved* *OMB No. 074-0188*

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
September 2010

3. REPORT TYPE AND DATES COVERED
September 2010

4. TITLE AND SUBTITLE
Triangular Mesh Generation of Elevation Versus Distance Course Profiles for use with Road Property Files

5. FUNDING NUMBERS

6. AUTHOR(S)
Michael Cole

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

U.S. Army Research, Development and Engineering Command
U.S. Army Tank- Automotive, Research, Development and Engineering Center
ATTN: RDTA-RS/MS157
Warren, MI 48397-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

21207

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved For Public Release; Distribution Unlimited.

12b. DISTRIBUTION CODE

A

13. ABSTRACT (Maximum 200 Words)

This report documents the process to convert right and left elevation versus distance course profiles into triangular meshes for use in Road Property Files (.rdf) used in RecurDyn multibody dynamics software. Firstly, MATLAB was used to combine the left and right elevation vs. distance course profile curves into one Importing Blend File. This file was imported into Pro/Engineer and used to create road geometry. The geometry was imported into HyperMesh and meshed, creating nodes and elements. The meshed geometry was exported as an ASCII mesh database file. MATLAB was used to parse the file and extract node and element data. Finally, the data was printed in the Road Property File (.rdf) format to be used in the RecurDyn multibody dynamics program.

14. SUBJECT TERMS
Road Property File (RDF), RecurDyn, distance vs elevation profile curve, mesh

15. NUMBER OF PAGES
16

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unclassified

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

Technical Report 21207
September 2010

Triangular Mesh Generation of Elevation Versus Distance Course Profiles for use with Road Property Files

Michael Cole
U.S. Army Research, Development and Engineering Command (RDECOM)
U.S. Army Tank-automotive and Armaments Research, Development and Engineering Center (TARDEC)
ATTN: RDTA-RS/MS157
6501 E 11 Mile Road
Warren, Michigan 48397-5000

1.0 INTRODUCTION

This report documents the process to convert right and left elevation versus distance course profiles into triangular meshes for use in Road Property Files (.rdf) used in RecurDyn multibody dynamics software. Multiple scripts were created in and processed with the MathWorks MATLAB technical computing program. Automation was used to help decrease conversion time and increase productivity.

The road or terrain on which a vehicle travels over must be defined to run a vehicle simulation. In this case, the simulation program (RecurDyn) required a specific file type and format be used when defining the road or terrain. To achieve this format, course profile data was required to be converted into the new file format (.rdf).

A brief overview of the conversion process is as follows:

- Left and right elevation vs. distance profile curves were combined into one file using MathWorks' MATLAB.
- Road geometry was created in PTC's Pro/Engineer Wildfire 5.0.
- Road geometry was meshed in Altair's HyperMesh.
- The mesh file was exported as an ASCII database file and imported into MATLAB for further processing.
- Scripts were created to scan the database file and extract, format, and output data in the Road Property File format.

2.0 ROAD GEOMETRY CREATION

Road Property Files require node and element data in order to correctly define the terrain surface (see Ref. 1). The node and element data comes from data originating with left and right elevation vs. distance profile curves. Combining both left and right curves was the first step in geometry creation. The geometry was generated in Pro/E to be used in Hypermesh to create nodes and elements which were necessary in the final Road Property file. The geometry was created by completing the following steps:

1. A script file was created in MATLAB to combine the left and right terrain course profile files into one Importing Blend File (.ibl) The MATLAB script is included in Appendix A.1. The elevation vs. distance course profile files simply contain two columns of data representing distance and elevation at that distance, respectively (see Figure 1). The following example displays the left column as distance in inches and the right column as elevation in inches.

0	-0.421332
3	-0.428685
6	-0.659071
9	-1.010741
12	-1.015889
15	-1.141309
18	-1.055657
21	-1.005185
24	-0.789697
27	-0.583433
30	-0.478471
33	-0.452257
36	-0.15139
39	-0.465199
42	-0.234502
45	-0.061961
48	-0.094372

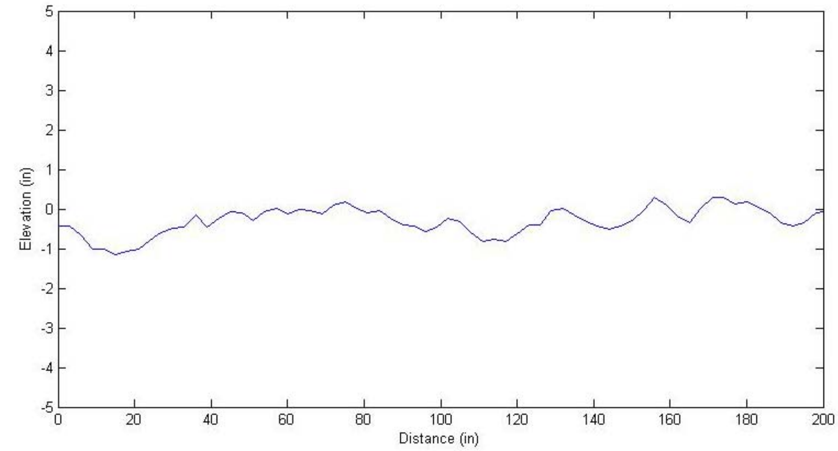


Figure 1 – Sample elevation vs. distance course profile curve file containing points representing distance in the left column and elevation at that distance in the right column (left) and a plotted elevation vs. distance curve (right).

2. The Importing Blend File contains points used by Pro/E to create geometry. The file defines one or more curves that are blended together in Pro/E to generate the surface geometry. The file must define the section and blend type as well as the sections and curves (see Figure 2 and Ref. 2). In MATLAB, a script loaded and stored the left and right elevation vs. distance profile values as matrices. Each matrix was printed in its own section in the Importing Blend File format to be used by Pro/E (see Ref. 2, Appendix A.2, and Figure 3).

Blend File Format

The imported blend data file, with the file extension .ibl, is in the following format.

Comments are contained between /* and */

```
/* beginning of file */

section_type     /* The section type (open or closed). */
blend_type       /* The blend method (arclength or pointwise).
              Both blend types require the same number of curve segments
              for each section. An arclength blend uses a general
              blending routine to connect the sections. The number of
              points in corresponding curves can be different for each
              section. A pointwise blend connects from point to point
              (point 1 in one curve connected to point 1 in the other
              curve). The corresponding curves in each section must have
              the same number of points.*/

begin section    /* Begin a new blend section.
              This appears at the beginning of each section. */

begin curve      /* Begin a new curve for the section.
              This appears at the beginning of each curve segment. */

1 x y z         /* The number is the point number (optional); X, Y,
2 x y z            Z are the coordinate values. */
.
.
.
# x y z
begin curve
1 x y z          /* The first point in this curve equals the last
2 x y z             point of the preceding curve. */
.
.
.
# x y z
/* end of file */
```

Figure 2 – Pro/ENGINEER Blend File Format help file describing correct Importing Blend File format.

```
open
pointwise
Begin section ! 1
Begin curve ! 1
0.000000    0.000000    -0.421332
0.000000    3.000000    -0.428685
0.000000    6.000000    -0.659071
0.000000    9.000000    -1.010741
.
.
.
0.000000    48123.000000    1.359609
```

Figure 3 – Sample Importing Blend File (.ibl) defining the section type, blend type, sections, and curves.

3. In Pro/Engineer (Pro/E), the Blend from File tool was used to import the Importing Blend File which created the geometry for the road's surface. This tool was opened by selecting: Insert > Advanced > Blend from File > Surface (see Figure 3). After starting the tool through the menu, selecting a coordinate system, selecting the Importing Blend File, and selecting the Material Side, Pro/E will automatically generate geometry by connecting the points in the Importing Blend File with a surface (see Figure 4). The geometry was then saved in Pro/Engineer as a Pro/E part file (.prt).

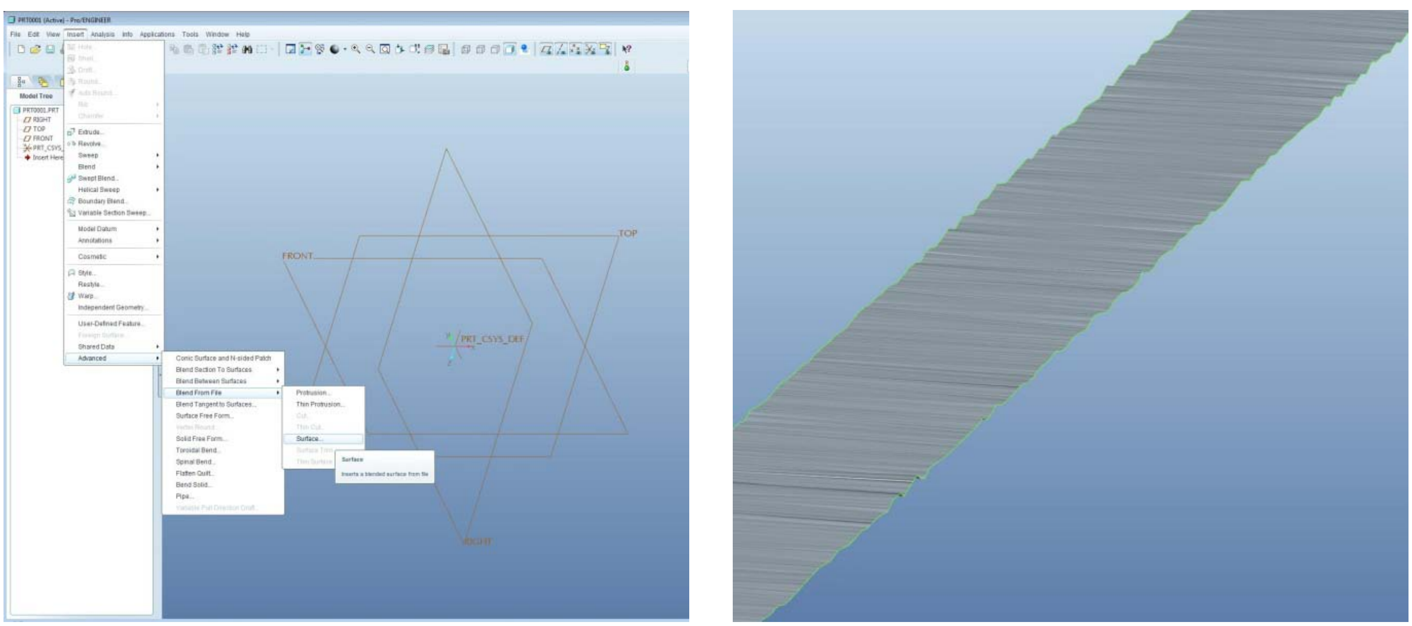


Figure 4 – Pro/E menu navigation for the Blend from File feature (left) and resulting road geometry created by Blend from File feature (right.

3.0 NODE AND ELEMENT CREATION

The Pro/E geometry was created to automate the process of node and element generation in HyperMesh. The Road Property File format requires that this node and element data be included in the file to define the road. The meshed geometry was exported to an ASCII mesh database file (see Ref. 3). The mesh database file containing node and element data was created by completing the following steps:

1. The road geometry created in Pro/Engineer was imported into HyperMesh. In HyperMesh, the geometry is imported by selecting: Import (green arrow) > Import Geometry (icon) > File Type: Auto Detect > Browse for file and selecting the road geometry Pro/E part file > Import (see Figure 5).

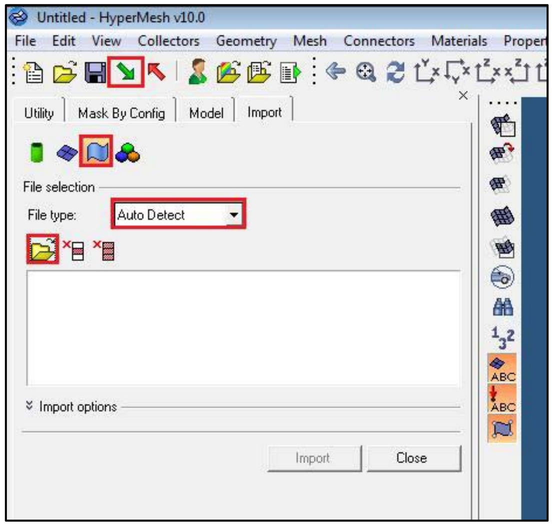


Figure 5 – HyperMesh menu navigation for importing geometry.

2. The automesh tool with triangular elements was used to mesh the surface and generate the nodes and elements. After importing the road surface geometry, meshing of the surface can begin. In the lower control panel, select: 2D > automesh > surfs (select road geometry) > mesh type: triads > mesh (see Figure 6). With these parameters set, HyperMesh will create nodes and triangular elements on the road geometry which can be exported to be used in MATLAB.

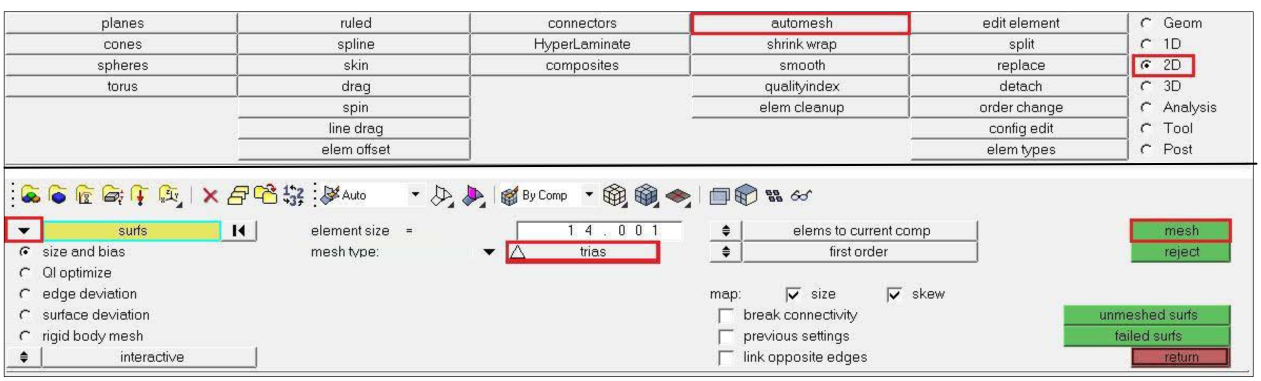


Figure 6 – HyperMesh menu navigation for meshing road surface geometry.

3. The resulting meshed road was exported by selecting: Export (red arrow) > Export FE Model (icon) > File Type: HMASCII > Template: HMASCII > File: name > Export (see Figure 7).

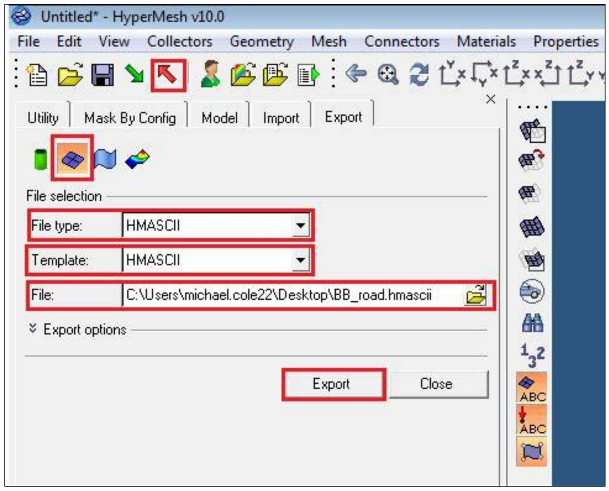
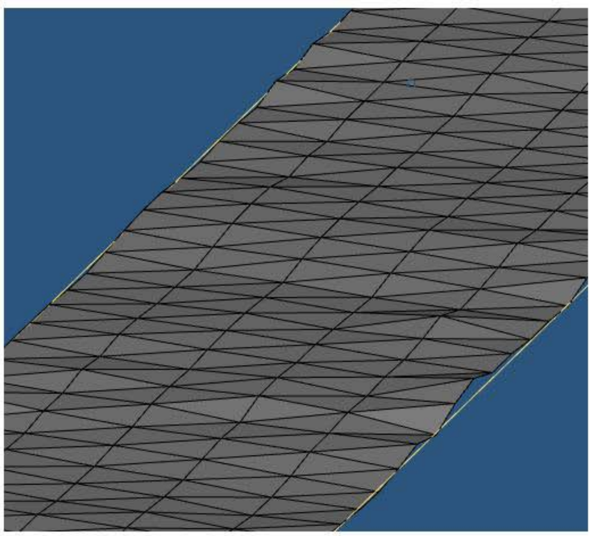


Figure 7 – HyperMesh menu navigation for exporting meshed surface as HMASCII text file.

4. The initial road geometry was exported as a HyperMesh ASCII mesh database file in text format (see Appendix A.3 and Figure 8). The text file allowed easy extraction of the node and element data to be used in the Road Property File.



```
8a0113b55e804ca78d8822d413dd181812ea317a790939a317e98922
e44f9f822

04d45adb23f749a2f4e8722970f6ab79a41baded117355cd51fcca73
0ab6698e0
       579ce3bf07fb3f731b821900
       END_OF_DATA
*endgeometry()
END GEOMETRY DATA

BEGIN CONTROLCARDS
END CONTROLCARDS

BEGIN SYSTEMS
END SYSTEMS

BEGIN NODES
*node(1,0.0      ,48123.0 ,1.359609,0,0,0,1,15)
*node(2,15.0     ,48123.0 ,1.385798,0,0,0,1,7)
*node(3,30.0     ,48123.0 ,1.411978,0,0,0,1,7)
*node(4,45.0     ,48123.0 ,1.438149,0,0,0,1,7)
*node(5,60.0     ,48123.0 ,1.46431 ,0,0,0,1,7)
*node(6,75.0     ,48123.0 ,1.490462,0,0,0,1,16)
*node(7,75.0     ,48108.92,1.496586,0,0,0,1,9)
*node(8,75.0     ,48094.83,1.479693,0,0,0,1,9)
*node(9,75.0     ,48080.75,1.253126,0,0,0,1,9)
*node(10,75.0     ,48066.67,1.100342,0,0,0,1,9)
*node(11,75.0     ,48052.59,0.897367,0,0,0,1,9)
*node(12,75.0     ,48038.51,0.612016,0,0,0,1,9)
```

Figure 8 – Meshed road geometry in HyperMesh displaying triangular elements (left) and a sample of the ASCII database file containing node and element data exported from HyperMesh (right).

4.0 MATLAB SCRIPTING

The technical computing program MATLAB was used to create the final Road Property file. Several script files were constructed to extract, organize, and output the Road Property file data from the HyperMesh ASCII database file. The following scripts were used:

1. A script file that parsed the HyperMesh ASCII mesh database file, extracting and storing the necessary node information including node numbers and the respective x, y, and z coordinates (see Appendix A.4).
2. A script file that parsed the HyperMesh ASCII mesh database file, extracting and storing the necessary element information including element numbers and the first, second, and third nodes which comprised each element (see Appendix A.5).
3. A script file that combined and printed all necessary information for the Road Property file in the Road Property file format (see Appendix A.6).

5.0 RESULTS

Figure 9 shows the Road Property File MATLAB output. Sections of node and element data were removed to shorten the text file for easier viewing.

```
!   ------------------
!   | ROAD DATA FILE |
!
!   Default road file - BB.rdf
!
METHOD
GENDATA
!
!  Conversion factors
!
X_SCALE
1.0
Y_SCALE
1.0
Z_SCALE
1.0
!
!Road origin is located at the following global coordinates in the
!data set.
!
ORIGIN
0.0 0.0 0.0
!
UP
0.0 1.0 0.0
!Road coordinate system is oriented with respect to the global origin
!by the following transformation matrix.
!
ORIENTATION
1  0 0
0  1 0
0  0 1
!
!Coordinates for the node points on road
!
NODES
20687
1  0.000000  48123.000000  1.359609
2  15.000000   48123.000000  1.385798
3  30.000000   48123.000000  1.411978
4  45.000000   48123.000000  1.438149
5  60.000000   48123.000000  1.464310
6  75.000000   48123.000000  1.490462
7  75.000000   48108.920000  1.496586
8  75.000000   48094.830000  1.479693
9  75.000000   48080.750000  1.253126
10 75.000000   48066.670000  1.100342
.
.
.
20687   30.022680   3037.347000   -0.848530
ELEMENTS
34469
1  1.000000  7.000000  6948.000000
2  1.000000  6947.000000   6948.000000
3  1.000000  6947.000000   6945.000000
4  1.000000  6948.000000   6946.000000
5  1.000000  9.000000  6940.000000
6  1.000000  8.000000  6949.000000
7  1.000000  6940.000000   6941.000000
8  1.000000  6949.000000   6944.000000
```

```
9   1.000000  6945.000000    6950.000000
10  1.000000  6946.000000    6953.000000
.
.
.
34469    1.000000  6948.000000    5.000000
```

Figure 9 – Road Property File used to define the road profile in the Redurdyn multibody dynamics program

6.0 SUMMARY/CONCLUSION

In order to simulate a vehicle's movement over certain terrain in RecurDyn, a Road Property file (RDF) that defines the terrain or road is required. The creation process of this file was partially automated to reduce conversion time and increase productivity.

The final RDF originated from elevation vs. distance course profile curves. These curves were used to create geometry in Pro/E that was meshed in HyperMesh (HM) to create nodes and elements. Using MATLAB, the node and element data was extracted from the HM exported ASCII mesh database file and printed in the resulting Road Property file format. Now, this file is ready to be used by the RecurDyn multibody dynamics program in conjunction with other data to simulate a vehicle travelling over a road surface or terrain.

7.0 APPENDIX

A.1 - MATLAB script to create Importing Blend File

```
%creates ibl file from two dist. vs. elevation curves

%load left curve and create x,y,z matrix
load bbL.crvdet
x = bbL;
xmat = [zeros([length(x),1]), x];

%load right curve and create x,y,z matrix, offset by 75 inches
load bbR.crvdet
y = bbR;
ymat = [75*ones([length(y),1]), y];

%create blend1.ibl file
fid = fopen('terrain.ibl', 'w');

%writes header info and x matrix
fprintf(fid, 'open\npointwise\nBegin section ! 1\nBegin curve ! 1\n');
fprintf(fid, '%7.6f\t%7.6f\t%7.6f\n', xmat');

%writes header info and y matrix
fprintf(fid, 'Begin section ! 2\nBegin curve ! 1\n');
fprintf(fid, '%7.6f\t%7.6f\t%7.6f\n', ymat');

fclose(fid);
```

A.2 - Resulting Importing Blend File sample

```
open
pointwise
Begin section ! 1
Begin curve ! 1
```

```
0.000000    0.000000    -0.421332
0.000000    3.000000    -0.428685
0.000000    6.000000    -0.659071
0.000000    9.000000    -1.010741
0.000000    12.000000   -1.015889
0.000000    15.000000   -1.141309
…
0.000000    48123.000000    1.359609
Begin section ! 2
Begin curve ! 1
75.000000   0.000000    -0.513719
75.000000   3.000000    -0.535514
75.000000   6.000000    -1.005605
75.000000   9.000000    -1.388840
75.000000   12.000000   -1.215158
75.000000   15.000000   -1.210530
75.000000   18.000000   -0.946456
75.000000   21.000000   -0.859729
75.000000   24.000000   -0.903716
75.000000   27.000000   -0.612990
…
75.000000   48123.000000    1.490537
```

A.3 - Sample ASCII mesh database file (exported from HyperMesh)

```
HYPERMESH Input Deck Generated by HyperMesh Version  : 10.0build60
Generated using HyperMesh-HmAscii Template Version : 10.0-SA1-140
*filetype(ASCII)
*version(10.0build60)

BEGIN DATA
BEGIN GEOMETRY DATA
*begingeometry()
      0000000000000000000014401f8b080000000000000b74bc0938d45ff43f3e635fb386b2562ada48
      2422271449911449a90845210945f67ddf97ac63ad1445916429a2148a90648b2c65df87313f7d7f
      5ff77efebfe7f9bf9f6732f3eacceb9e73eebdaf73ee7b065ac2ffbd58da326c9559c4818d404ba0
      5de0e4f50d3c7b7daf360facfd4c4cf877bd5209211208bbe30862ffdeb571f5b1fa92b02be07f9f
      acfe201068fef7b1f31f4abbfaf86742bc214420c8ad3ef977fd03fe7bb1adbe505535d1d339acbe
      47574f1fff9722e3ff7dfeef0dff98b67bacfec3bbfa409474ab2ffe5df4ab8f7fcfc53d0844ce7f
      2f9005d3ffbe58fd4160597dfcfbcf6d1e04e0fc5fee7ff8aa39e3fb1e07a512b3ca70b9a1d29ceb
      833042160f2a0d36aba4ff870bdc84ea7f3fa207e14b8ca0fed8a01930acbe69ab07137072fc7bf7
      bfc7bf08d7dc597dfdff5cd4d5cbc46d10cc42072ecc56ceaafc1bfddffb093cff8fe17f5e3a597c
      38f1ef3dfea65207e2ab6755fefdd7bf71ff5dffc2fa9ff139579ffc4bf9ffcff8947f0357e8c119
…
BEGIN NODES
*node(1,0.0     ,48123.0 ,1.359609,0,0,0,1,15)
*node(2,15.0    ,48123.0 ,1.385798,0,0,0,1,7)
*node(3,30.0    ,48123.0 ,1.411978,0,0,0,1,7)
*node(4,45.0    ,48123.0 ,1.438149,0,0,0,1,7)
*node(5,60.0    ,48123.0 ,1.46431 ,0,0,0,1,7)
*node(6,75.0    ,48123.0 ,1.490462,0,0,0,1,16)
*node(7,75.0    ,48108.92,1.496586,0,0,0,1,9)
*node(8,75.0    ,48094.83,1.479693,0,0,0,1,9)
*node(9,75.0    ,48080.75,1.253126,0,0,0,1,9)
*node(10,75.0    ,48066.67,1.100342,0,0,0,1,9)
*node(11,75.0    ,48052.59,0.897367,0,0,0,1,9)
*node(12,75.0    ,48038.51,0.612016,0,0,0,1,9)
*node(13,75.0    ,48024.43,0.526117,0,0,0,1,9)
…
BEGIN COMPONENTS
```

```
*component(1,"BB_ROAD.PRT",0,58,0)
*altsurface(1,1,20)
*component(2,"auto1",0,11,0)
*tria3(1,1,7,6948,6,0)
*tria3(2,1,6947,6948,7,0)
*tria3(3,1,6947,6945,6946,0)
*tria3(4,1,6948,6946,4,0)
*tria3(5,1,9,6940,6949,0)
*tria3(6,1,8,6949,6947,0)
*tria3(7,1,6940,6941,6944,0)
*tria3(8,1,6949,6944,6945,0)
*tria3(9,1,6945,6950,6952,0)
*tria3(10,1,6946,6953,4,0)
```

A.4 - MATLAB script to extract node data from ASCII file

```
%this script will parse the .hmascii file and extract each node
%number, x, y, and z coordinates

fid = fopen('BB_road.hmascii', 'rt');
fido = fopen('nodesall.txt', 'wt');

%scans each line of BB_road.hmascii and checks against string: *node(
%prints any lines that match above case to nodesall.txt
while 1;
    tline = fgetl(fid);
    if tline==-1, break, end;
    if  strncmp(tline, '*node(', 6);
        fprintf(fido,'%s\n', tline);
    end;
    disp(tline);
end
fclose(fid);
fclose(fido);

%reads nodesall.txt and extracts the node number, x, y, z coordinates
[node, x, y, z] = textread('nodesall.txt', '*node(%d %f %f %f %*[^\n]', 'delimiter',
',');

%creates nodeesformat.txt file to store formatted nodal information
%prints: total number of nodes = length(node)
%prints variables: node, x, y, z
nodesmat = [node, x, y, z];
fid2 = fopen('nodesformat.txt', 'wt');
fprintf(fid2, '%f\n', length(node));
fprintf(fid2, '%f\t %f\t %f\t %f\n', nodesmat');
fclose(fid2);
```

A.5 - MATLAB script to extract element data from ASCII file

```
%this script will parse the .hmascii file and extract each element
%number, node 1, node 2, node 3

fid = fopen('BB_road.hmascii', 'rt');
fido = fopen('elemall.txt', 'wt');

%scans each line of BB_road.hmascii and checks against string: *tria3(
%prints any lines that match above case to elemall.txt
while 1;
    tline = fgetl(fid);
```

```
    if tline==-1, break, end;
    if  strncmp(tline, '*tria3(', 7);
        fprintf(fido,'%s\n', tline);
    end;
    disp(tline);
end
fclose(fid);
fclose(fido);

%reads elemall.txt and extracts the element number & nodes 1, 2, 3
[elem, n1, n2, n3] = textread('elemall.txt', '*tria3(%d %f %f %f %*[^\n]', 'delimiter',
',');

%creates elemformat.txt file to store formatted nodal information
%prints: total number of elements = length(elem)
%prints variables: node, x, y, z
elemmat = [elem, n1, n2, n3,];
fid2 = fopen('elemformat.txt', 'wt');
fprintf(fid2, '%f\n', length(elem));
fprintf(fid2, '%f\t %f\t %f\t %f\t %f\t %f\n', elemmat');
fclose(fid2);
```

A.6 - MATLAB script that prints extracted data in Road Property File Format

```
%this script outputs road profile data in .rdf format

%define conversion factor
x_scale = 1.0;
y_scale = 1.0;
z_scale = 1.0;

%define origin
origin_x = 0;
origin_y = 0;
origin_z = 0;

%define direction for up
up_x = 0;
up_y = 1;
up_z = 0;

%define orientation
O = [1 0 0; 0 1 0; 0 0 1];

%begin printing text
fid1 = fopen('crvdet2rdf.txt', 'wt');

fprintf(fid1, '\r!   ------------------\n!   | ROAD DATA FILE |\n!\n!   Default road file
- BB.rdf\n!\n');

fprintf(fid1, 'METHOD\nGENDATA\n');

fprintf(fid1, '!\n!  Conversion factors\n!\n');

fprintf(fid1, 'X_SCALE\n%2.1f\n', x_scale);
fprintf(fid1, 'Y_SCALE\n%2.1f\n', y_scale);
fprintf(fid1, 'Z_SCALE\n%2.1f\n', z_scale);
```

```
fprintf(fid1, '!\n!Road origin is located at the following global coordinates in
the\n!data set.\n!\n');

fprintf(fid1, 'ORIGIN\n%2.1f %2.1f %2.1f\n!\n', origin_x, origin_y, origin_z);

fprintf(fid1, 'UP\n%2.1f %2.1f %2.1f\n', up_x, up_y, up_z);

fprintf(fid1, '!Road coordinate system is oriented with respect to the global origin\n!by
the following transformation matrix.\n!\n');

fprintf(fid1, 'ORIENTATION\n');
fprintf(fid1, '%d\t %d\t %d\n', O);

%print list of nodes generated from nod.m
fprintf(fid1, '!\n!Coordinates for the node points on road\n!\n');
fprintf(fid1, 'NODES\n%d\n', length(node));
fprintf(fid1, '%d\t %f\t %f\t %f\n', nodesmat');

%print list of elements generated from ele.m
fprintf(fid1, '\rELEMENTS\n%d\n', length(elem));
fprintf(fid1, '%d\t %f\t %f\t %f\n', elemmat');
```

CONTACT

The author is an engineer with the U.S. Army Research, Development and Engineering Command (RDECOM), located at the U.S. Army Tank-automotive and Armaments Research, Development and Engineering Center (TARDEC). Interested parties can contact him at the U.S. Army Tank-automotive and Armaments Research, Development and Engineering Center (TARDEC), ATTN: RDTA-RS/MS157, 6501 E 11 Mile Rd., Warren, Michigan 48397-5000, email: “michael.cole22@us.army.mil”.

REFERENCES

1. FunctionBay Recurdyn Help Doncumentation
   Road Property File Format
   Help > Appendix & Reference > Appendix B > Road Property File Format

2. PTC Pro/Engineer Help documentation
   Blend File Format
   Help > Pro/E Functional Areas > Part Modeling > Part Modeling > Base Features > Blends > Blend File Format
   Example: Importing Blend File
   Help > Pro/E Functional Areas > Part Modeling > Part Modeling > Base Features > Blends > Example: Importing Blend File

3. Altair HyperMesh Help documentation
   Input Format
   Help > HyperMesh > Refereence Guide > HMASCII > Input Format
   *node()
   Help > HyperMesh > Refereence Guide > HMASCII >Commands and Functions > HMASCII Commands > *node()
   *tria3()
   Help > HyperMesh > Refereence Guide > HMASCII >Commands and Functions > HMASCII Commands > *tria3()

4. MathWorks MATLAB Help documentation
   Type in command window:
   fprintf
   fopen
   fid
   strncmp

5. MathWorks.com Help documentation
   Textscan – http://www.mathworks.com/help/techdoc/ref/textscan.html

DEFINITIONS, ACRONYMS, ABBREVIATIONS

RDECOM – U.S. Army Research, Development and Engineering Center
TACOM - U.S. Army Tank-automotive and Armaments Command
TARDEC - TACOM Research, Development and Engineering Center
PRO/E – PTC Pro/Engineer
HM – Altair HyperMesh
MATLAB – MathWorks Matrix Laboratory
RDF – Road Property File (FunctionBay RecurDyn)
ASCII – American Standard Code for Information Interchange

DISTRIBUTION

The distribution statement “STATEMENT A; Approved for public release; distribution is unlimited.